



SPECIFICATION

08 353008

Docket No. AT9-93-110

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN THAT I, **TIMOTHY MICHAEL SKERGAN**, a citizen of the United States of America residing in the State of Texas, have invented new and useful improvements in

A METHOD AND SYSTEM FOR MANIPULATING
A PLURALITY OF GRAPHICAL POINTERS

of which the following is a specification:



BACKGROUND OF THE INVENTION

08 353008

1. Technical Field:

The present invention relates in general to an improved method and system for data processing, and in particular to an improved method and system of user interface to a data processing system. Still more particularly, the present invention relates to an improved method and system within a data processing system for manipulating a plurality of graphical pointers utilizing a single graphical pointing device.

2. Description of the Related Art:

Within data processing systems, user interface is accomplished in a variety of ways. An increasingly common method of user interface is a graphical user interface (GUI), which provides a user with a graphical and intuitive display of information.

The technique universally used in data processing systems to interact with a GUI display is to display a single graphical pointer within the display device, which a user may control with a graphical pointing device, such as a mouse, track ball, or joystick. Utilizing the graphical pointing device, a user may select a point at the position indicated by the graphical pointer by depressing a button associated with the graphical pointing device. The response of the data processing system to the user's selection is determined by the application or operating system software. For example, in some data processing systems, if the point selected is within an object or icon displayed within the display device, the user can relocate the object or icon utilizing the drag-and-drop technique well-known in the art. Conventionally, the motion of the graphical pointer within the GUI display is

1 linearly related to the movement of the graphical pointing device. For
2 example, if a user moves the graphical pointing device 2 units to the right, the
3 graphical pointer responds by moving 2 scaled units to the right within the
4 display.

5

6 Word processing programs and other similar applications often
7 provide a cursor in addition to the graphical pointer. Utilizing the graphical
8 pointing device, a user may position the cursor at locations within the display
9 by clicking on a mouse button. Cursors and other indicia of position are
10 distinguished from graphical pointers in that they do not move in conjunction
11 with the graphical pointer in response to operation of the graphical pointing
12 device.

13

14 Although utilizing a graphical pointer provides an intuitive
15 method of user interaction with a GUI display, it is often inefficient, particularly
16 if the display consists of multiple windows, or a single display which is large
17 or complex. For example, if a display includes five separate windows in
18 which a user is manipulating objects, a large portion of the time required to
19 complete a task is spent traversing the display with the graphical pointer.
20 The inefficiency is greatest when the task requires repetitive switching
21 between the different windows.

22

23 Consequently, it would be desirable to have an improved
24 method and system for manipulating a plurality of graphical pointers within a
25 display utilizing a single graphical pointing device.

SUMMARY OF THE INVENTION

It is therefore one object of the present invention to provide an improved method and system for data processing.

It is another object of the present invention to provide an improved method and system for user interface to a data processing system.

It is yet another object of the present invention to provide an improved method and system within a data processing system for manipulating a plurality of graphical pointers utilizing a single graphical pointing device.

The foregoing objects are achieved as is now described. An improved method and system for manipulating a plurality of graphical pointers utilizing a single graphical pointing device are disclosed. A plurality of graphical pointers are displayed within a display device. A user may then temporarily select one graphical pointer among the plurality of graphical pointers. During the selection, the selected graphical pointer is manipulated in response to operation of a single graphical pointing device. A point within the display device specified by the position of the selected graphical pointer is selected in response to closure of a switch associated with the selected graphical pointer.

1 **BRIEF DESCRIPTION OF THE DRAWINGS**

2
3 The novel features believed characteristic of the invention are
4 set forth in the appended claims. The invention itself, however, as well as a
5 preferred mode of use, further objectives and advantages thereof, will best
6 be understood by reference to the following detailed description of an
7 illustrative embodiment when read in conjunction with the accompanying
8 drawings, wherein:

9
10 **Figure 1** illustrates a data processing system utilizing the
11 method and system of the present invention;

12
13 **Figure 2** depicts a window utilized to manage multiple graphical
14 pointers within a data processing system;

15
16 **Figure 3** illustrates the operation of the window depicted in
17 **Figure 2**;

18
19 **Figure 4** depicts two graphical pointers moving in a fixed
20 relationship in response to the operation of a graphical pointing device;

21
22 **Figure 5** illustrates a three-dimensional display in which the
23 relative motion of two graphical pointers is governed by a mathematical
24 function; and

25
26 **Figures 6A and 6B** together form a flowchart of the process
27 utilized by a data processing system to manage multiple graphical pointers.

1 DETAILED DESCRIPTION OF PREFERRED EMBODIMENT

2
3 With reference now to the figures and in particular with
4 reference to **Figure 1**, there is illustrated a data processing system
5 implementing the present invention. Data processing system **10** comprises
6 display device **12**, processor system **14**, keyboard **16**, and mouse **18**. In a
7 manner well-known in the art, a user inputs data to processor system **14**
8 utilizing keyboard **16**. Processor system **14** outputs data to a user via display
9 device **12**.

10
11 As illustrated, mouse **18** includes five mouse buttons, rather
12 than two, as is common in prior art systems. Each of the five buttons of
13 mouse **18** corresponds to one of graphical pointers **20-28**, displayed within
14 display device **12**. Utilizing mouse **18**, the user may determine the positions
15 of graphical pointers **20-28** within display device **12**. A user selects a point
16 within the display by depressing the button among mouse buttons **19** which
17 corresponds to the graphical pointer among graphical pointers **20-28** which
18 is positioned at the selected point.

} each button
↔ one graphic
pointer

19
20 In the present invention, graphical pointers may be distinguished
21 in a variety of ways. As depicted in **Figure 1**, graphical pointers **20-26** are
22 distinguished from graphical pointer **28**, which is a small transparent circle.
23 Characteristics distinguishing graphical pointers include the color, size, shape,
24 and direction in which the graphical pointer points.

25
26 Referring now to **Figure 2**, there is depicted a window utilized
27 to manage multiple graphical pointers. The creation and management of
28 multiple graphical pointers could occur at three locations within a data
29 processing system. First, and most simply, a fixed number of graphical
30 pointers could be incorporated into the design of the hardware and operating

1 system of the data processing system. Second, graphical pointers could be
2 created and managed utilizing an application program. Third, as illustrated
3 in **Figure 2**, multiple graphical pointers could be created and managed by a
4 window manager in windowing software.

5

6 As depicted, window **32** is displayed within display **30**. A user
7 opens window **32** through invocation of the program "Mouse Manager" by
8 selecting the program title on display menu bar **34**. Window **32** includes
9 menu bar **36**, pointer status box **38**, and sequence box **40**.

10

11 Utilizing an active graphical pointer or other input device, such
12 as a speech interpreter or the keyboard, a user may select one of the five
13 menu items displayed within menu bar **36**. When a menu item is selected,
14 a pull-down box of functions is displayed beneath the selected menu item.
15 As illustrated in **Figure 3**, when a user selects "Add" on menu bar **36**, the
16 user is presented with pull-down box **42**, containing of options which enable
17 the user to add additional graphical pointers, as well as to select the shape,
18 size, color, and other attributes of the displayed graphical pointers. In
19 addition, pull-down box **42** includes a function which allows the user to define
20 a "Home" window and "Home" position for a given pointer. When a graphical
21 pointer having a home window and home position is activated, the graphical
22 pointer is reset to the specified position in the home window. In general, the
23 home position is specified by the X-Y coordinates of the home position with
24 respect to the upper left-hand corner of the home window. However, in
25 windowing environments supporting three-dimensional displays, the home
26 position is specified by X-Y-Z coordinates.

27

28 Within menu bar **36**, menu item "Delete" is also provided.
29 Utilizing the functions displayed within the pull-down box associated with
30 "Delete" a user may delete a selected graphical pointer.

1 Functions associated with the menu item "Create_Group" enable
2 a user to select a subset of the multiple graphical pointers which may be
3 manipulated simultaneously by movement of a mouse or other graphical
4 pointing device. These subsets or "Groups" of active pointers may be
5 ordered in a sequence determined by the user. Graphical pointers which are
6 not active are not displayed in order to prevent confusion and the obscuring
7 of displayed objects.

8

9 To select an active group among the defined groups, a user
10 presses a hot-key, which the user defines utilizing the pull-down menu
11 associated with the menu item "Hot-Keys". The hot-key may be any
12 combination of keystrokes or other inputs which do not have conflicting
13 definitions in other contexts. When the user presses the hot-key while in an
14 application, the mouse manager program is invoked. At that time, a user
15 may select the next group in the user-defined sequences of groups, which is
16 the default selection, or may select another group in the list. After the user
17 has selected the desired active group, the user presses enter to return to the
18 active application.

19

20 A user can also define the relative motions of the graphical
21 pointers utilizing menu item "Create_Link". Selection of Create_Link enables
22 a user to define the motion of a graphical pointer relative to the input from the
23 graphical pointing device utilizing a mathematical function. The function could
24 be a one-dimensional linear relationship or another more complex function.

25

26 Within pointer status box 38 information about the multiple
27 graphical pointers is displayed in table format. As illustrated in pointer status
28 box 38, the system currently has two graphical pointers, which are both
29 active, meaning that both move in response to operation of the graphical
30 pointing device.

1 The third column "HOME" indicates each graphical pointer's
2 home window, home position, and reset/offset. As indicated, graphical
3 pointer 1 has a home position of 4.25 units in the X direction and 2.75 units
4 in the Y direction from the upper left-hand corner of window one. In addition,
5 graphical pointer 1 has reset/offset (Δ) of 2 units in the Y direction, which
6 means that the home position of graphical pointer 1 is advanced 2 units in
7 the Y direction when graphical pointer 1 is reset to its home position by
8 selecting a group in which pointer 1 is active. The remaining three columns
9 of the table displayed within pointer status box 38 indicate that the user has
10 created three groups of graphical pointers. Group1 contains only graphical
11 pointer 1. Similarly, Group3 contains only graphical pointer 2. Group2
12 includes both graphical pointer 1 and graphical pointer 2. Below the group
13 name, the mathematical relationship governing the motion of each pointer is
14 listed. The equations "1 - x" and "1 - y" indicate that the motion of the
15 graphical pointers within the display will directly correspond the movement of
16 the graphical pointing device.

17

18 Below pointer status box 38, sequence box 40 is displayed.
19 Sequence box 40 indicates to a user the default sequence in which the
20 groups will be selected, and the currently defined hot-key sequence which the
21 user utilized to invoke the mouse manager window.

22

23 With reference now to **Figure 4**, there is illustrated a preferred
24 embodiment of the present invention in which two graphical pointers are
25 displayed, with each graphical pointer resident within its own window.
26 Graphical pointers 54 and 58 correspond to graphical pointers 1 and 2,
27 respectively, which were described with reference to **Figure 2**. As indicated
28 by the dashed-line arrows, graphical pointers 54 and 58 both move linearly
29 in the Y direction in response to operation of the graphical pointing device to
30 the positions indicated at reference numerals 54a and 58a, respectively.

1 Because a user has selected object 56 utilizing graphical pointer 54,
2 object 56 moves in conjunction with graphical pointer 54 in the familiar drop-
3 and-drag manner.

4
5 If the user pressed the hot-key sequence, which was defined as
6 Cntl-Shift-Z, and selected the default group of graphical pointers, graphical
7 pointer 58 would become inactive, since Group1, the subsequent group in the
8 sequence, contains only graphical pointer 1. When graphical pointer 58
9 again becomes active, the position of graphical pointer 58 will be reset to the
10 home position indicated within pointer status box 38 of **Figure 2**. The home
11 position of graphical pointers in two-dimensional displays is determined with
12 respect to the upper left-hand corner of the home window of the graphical
13 pointer, which has coordinates of (0, 0).

14
15 Referring now to **Figure 5**, there is depicted a preferred
16 embodiment of the present invention in which multiple pointers are displayed
17 within a three-dimensional display. Within display 30, graphical pointers 70
18 and 72 are displayed, in conjunction with cube 74. As indicated by the
19 dashed-line illustration of arrows 76 and 78, graphical pointers 70 and 72 are
20 both active, and move to the positions indicated at reference numerals 70a
21 and 72a in response to operation of a graphical pointing device. However,
22 unlike graphical pointer 70, the movement of graphical pointer 72 is not
23 linearly related to the operation of the graphical pointing device. Instead, its
24 movement is determined by a three-dimensional user-defined mathematical
25 function. Thus, even though the display is only two-dimensional, a three-
26 dimensional coordinate system may be utilized.

27
28 Referring now to **Figures 6A and 6B**, there is depicted a
29 flowchart of the process utilized by the present invention to manage multiple
30 graphical pointers. Because of the wide variety of possible methods to

1 implement multiple graphical pointers and the divergence between window
2 manager programs, processes utilized to manage multiple graphical pointers
3 will vary between data processing systems. The process illustrated in
4 **Figures 6A and 6B** describes a preferred embodiment of the present
5 invention implemented in an X-Windows windowing environment.

6
7 In an X-Windows environment, data utilized to display a single
8 graphical pointer is stored within a data structure in memory. To implement
9 multiple graphical pointers according to the present invention, the single data
10 structure is replaced by a plurality of data structures which are maintained in
11 memory by the window manager software utilizing a list of software pointers.
12 The window manager software and active application programs communicate
13 utilizing an asynchronous protocol. Utilizing this protocol, the window
14 manager software sends active application programs messages when
15 graphical pointers generate events relevant to that application. For example,
16 a message is sent to an application program by the window manager
17 software when a user selects an object within the application's window
18 utilizing a graphical pointer.

19
20 As illustrated, the process begins in block **100**, and thereafter
21 proceeds to block **102**, which illustrates the window manager program polling
22 to determine if the user has moved the mouse, or other graphical pointing
23 device interfaced to the data processing system. If the mouse has not
24 moved, the process proceeds to block **114**. However, if the mouse has
25 moved, the process proceeds to block **104**, which depicts accessing the data
26 structure of the first graphical pointer utilizing a software pointer to the
27 graphical pointer's data structure. Thereafter, the process proceeds to block
28 **106**, which illustrates determining if the graphical pointer is currently active.
29 If the graphical pointer is not active, the process returns to block **104**, which

1 depicts accessing the data structure of the graphical pointer whose
2 associated software pointer is next in the list of software pointers.

3

4 If, however, the graphical pointer is active, the process
5 proceeds from block **106** to block **108**, which illustrates computing the
6 coordinates at which to display the graphical pointer. To compute the display
7 coordinates of the graphical pointer, the window manager software applies
8 either a default equation, which is an essentially linear relationship between
9 the mouse and the graphical pointer, or a user-defined equation, which the
10 user determines utilizing the mouse manager software. The process then
11 proceeds to block **110**, which illustrates displaying the graphical pointer at the
12 calculated location. Thereafter, the process proceeds to block **112**, which
13 illustrates determining if there are more graphical pointers, as indicated by
14 additional software pointers within the list. If more graphical pointers remain
15 to be processed, the process returns from block **112** to block **104**. If all
16 active pointers have been repositioned in response to the mouse movement,
17 the process proceeds from block **112** to block **114**, which illustrates sending
18 messages generated by the motion of the graphical pointers to the interested
19 applications (i.e., application for which the new locations of the graphical
20 pointers created events).

21

22 The process then proceeds to block **116**, which depicts
23 determining if a mouse button associated with an active graphical pointer has
24 been pressed. If a mouse button has been pressed, the process proceeds
25 to block **118**, which depicts sending a message to the application for which
26 the selection created an event. The process then proceeds from either block
27 **116** or block **118** to block **120**.

28

29 Block **120** illustrates determining if the hot-key has been
30 pressed. As described above, the hot-key is a user-defined series of

1 keystrokes, which are utilized when selecting a new group of active graphical
2 pointers. If the hot-key has not been pressed, the process proceeds from
3 block **120** to block **136**, where it terminates. If the hot-key has been pressed,
4 the process proceeds to block **122**, which illustrates recording which
5 application program is currently active. The process then proceeds to
6 block **124**, which depicts activating the mouse manager program and
7 displaying the mouse manager window. As described above, the next group
8 of active graphical pointers is set by default to be the group following the
9 currently active group in the user-defined sequence. However, the user may
10 select any of the defined groups utilizing the arrow keys provided on the
11 keyboard or other input device.

12

13 Thereafter, the process proceeds to block **126**, which depicts
14 determining if a new group of graphical pointers was selected. If a new
15 group was selected, the process proceeds to blocks **128** and **130**, which
16 illustrate deactivating the current group of graphical pointers and activating
17 the newly selected group of graphical pointers by building a new list of
18 software pointers to the data structures for the newly selected graphical
19 pointers. This "currently active" list is a subset of the list of all software
20 pointers which point to all of the graphical pointer data structures. As
21 described above, inactive graphical pointers are not displayed within the
22 display device of the data processing system.

23

24 Thereafter, the process proceeds to block **132**, which illustrates
25 closing the mouse manager window and returning to the active application.
26 The process then proceeds to block **134**, which depicts sending messages,
27 for example, the locations of the active graphical pointers, to the active
28 application. Thereafter, the process terminates at block **136**.

29

1 While the invention has been particularly shown and described
2 with reference to a preferred embodiment, it will be understood by those
3 skilled in the art that various changes in form and detail may be made therein
4 without departing from the spirit and scope of the invention.